Stacking.

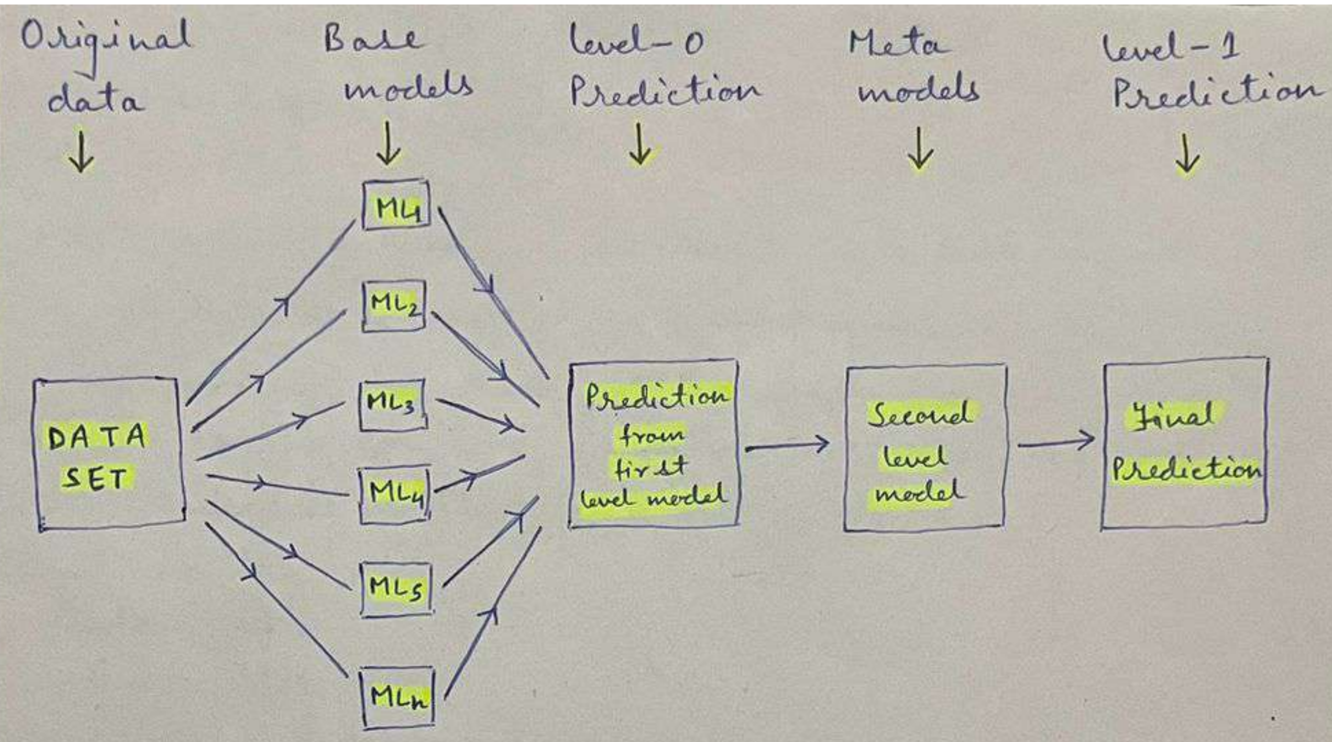
Ensemble technique where numerous weak learners are ensembled in parallel manner in such way that by combining them with meta learners, we can predict better predictions for the future.

Stacking is also known as stacked generaliza-tion. All sub-models equally participate as per their performance weights and build new model with better prediction.

Weak learners are ensembled parallely same as case of Bagging but it differ in output prediction w.r.t case in Bagging.

Architecture of stacking:

Architecture is designed in such a way that it consists of two or more base/learner's models and a meta model that combines the predictions of the base models. Base model are called level 0 models and the meta model is level 1 model. Thus is consist of original data, Primary level models, Primary level prediction, secondary level model and final prediction.



Original data : This data is divided into n-folds and is considered test/training data.

Base models : level-0 models that uses test/traning data to pravide compiled prediction.

Level-0 prediction : Each base model is triggered on same data and pravides different predictions and called level-0 prediction.

Meta Model : Help best combine the predictions of base models and is called level-1 model.

Level-1 prediction : Meta model learn how to best combine predictions of base models and is trained on different predictions mode by individual base models.

Example with calculation: [Ensemble Tech]

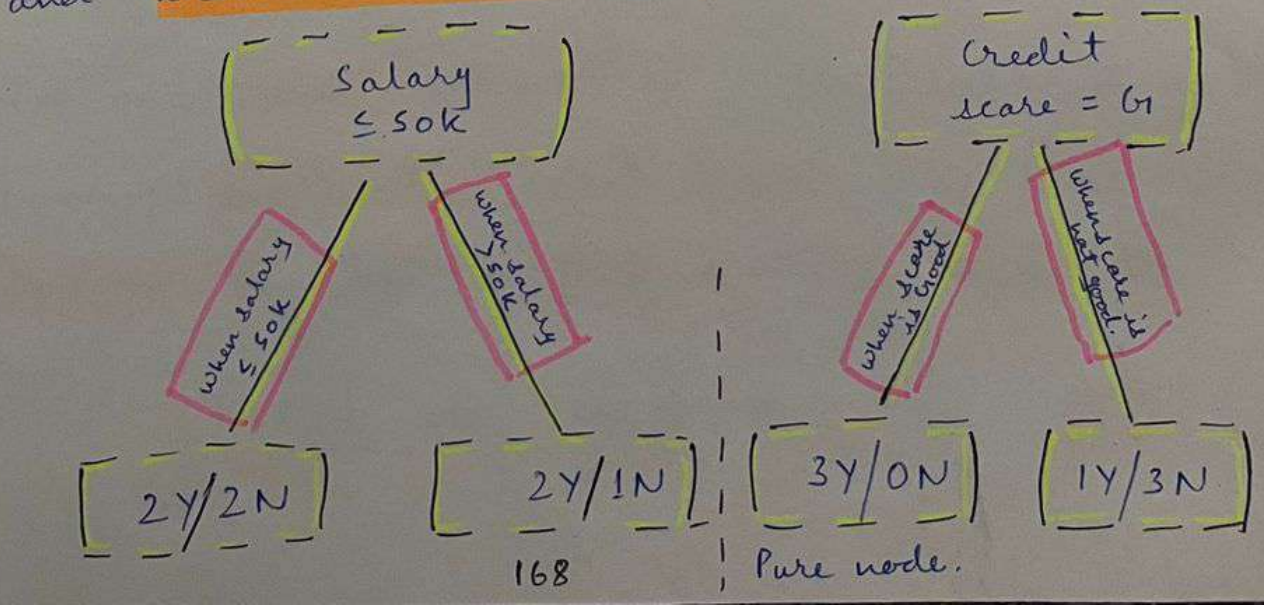
Consider below data: | Using Ada boosting.

Salary	Credit Sc.	Approval	Weight
<= 50k	B	No	1/7
<= 50k	G	Yes	1/7
<= 50k	G	Yes	1/7
> 50k	B	No	1/7
> 50k	G	Yes	1/7
> 50k	N	Yes	1/7
<= 50k	N	No	1/7

B : BAD   N: Normal   G: Good

As we have seven data points thus we have provided 1/7 as initial weight.

Drawing decision Tree based on Salary ≤ 50k and credit score = Good.



we need to calculate Entrapy or Gini - coef. for bath the stumps and one having max. one we will go with it.

As the stum of Credit scare has a pure node so calculation is higher for it.

Thus, we will consider credit scare for further calculation.

Salary	Credit Sc.	Approval	weight.
< = 50k	B	NO	1/7
< = 50k	G	Yes	1/7
< = 50k	G	Yes	1/7
> 50k	B	NO	1/7
> 50k	G	Yes	1/7
> 50 k	N	Yes	1/7
<= 50k	N	NO	1/7

For one row approval is given instead of having Narmal credit scare. Thus that datapoint will be passed to further W.L

Tatal error as of now = 1/7 [T. E]

we will calculate further weight to update:

$$\text{Performance of stump} = \boxed{\frac{1}{2} \ln\left[\frac{1-T.E}{T.E}\right]}$$

$$= \frac{1}{2} \ln(6) \approx 0.896$$

Now,

weight of first w.L (α_1) = 0.896

Now,

weight of carrect data points will be

$$\text{updated one} = w_{new} = \boxed{w_{old} \times e^{-\alpha_1}}$$

$$= \frac{1}{7} \times e^{-0.896} = 0.058$$

Weight of wrong data points will be

$$w_{new} = \boxed{w_{old} \times e^{\alpha_1}}$$

$$= \frac{1}{7} \times e^{0.896} = 0.349$$

Updating New weights.

Salary	Credit Sc.	Approval	w_{old}	w_{new}
<= 50k	B	NO	1/7	.058
<= 50k	G	Yes	1/7	.058
<= 50k	G	Yes	1/7	.058
> 50k	B	NO	1/7	.058
> 50k	G	Yes	1/7	.058
<= 50k	N	NO	1/7	.058
> 50k	N	Yes	1/7	.349

As the sum of weight earlier was 1. So we need to standardize the updated weight as well.

Normalizing and assigning bins.

$\div .697$

Salary	Credit Sc	Approval	weight	W_{new}	Normalized weight
<=50k	B	No	1/7	.058	.08
<=50k	G	Yes	1/7	.058	.08
<=50k	G	Yes	1/7	.058	.08
>50k	B	No	1/7	.058	.08
>50k	G	Yes	1/7	.058	.08
>50k	N	Yes	1/7	.349	.5
<=50k	N	No	1/7	.058	.08
				Sum : .697	≈ 1

Now, dividing updated weight with it's total column sum to Normalize value b/w 0 — 1.

Normalized weight	Bin Assignment
.08	0 — .08
.08	.08 — .16
.08	.16 — .24
.08	.24 — .32
.08	.32 — .4
.5	.4 — .9
.08	.9 — 1

Thus, now we have the bin size & max is of one having wrong data point.

Now, most of the weight data points will lies within that bin range [wrong one]

Then if predicted wrong data will further pass on to the further ML algo. Again if it continue wrong prediction we continue on increasing weight by using [$\text{weight}_{old} \times e^{\pm \alpha_i}$] and decreasing for right one.

Lastly we will have weightage sum and we will take final output for yes/NO accordingly whose weightage will be more.

In code we will do so by adjusting n-estimators to get best accuracy.

Even we can use combinations of base learners to have better accuracy in some cases. Ex - S.V.C, S.V.M.